User Interface Architecture

The invention provides, in still further aspects, an improved architecture of a digital data processor of the type used in a storage area network (SAN). The digital data processor, which

5   can be the aforementioned manager digital data processor, executes a process, herein referred to as a manager process, to maintain a representation of the SAN topology or at least an attribute thereof. A graphical output device displays the SAN representation. A further process, herein referred to as a user interface process, controls the output device for purposes of displaying that representation. An interface element, residing on the manager digital data processor or another

10   data processor, effects retrieval of the SAN representation, for example, in response to a request from the user interface process. It transmits that representation to the user interface process for display on the graphical output device.

In a related aspect, the invention provides a SAN as described above in which the interface

15   element includes a requester that receives a request from the user interface process for retrieval of the SAN representation from the manager process. For example, the user interface process can transmit such a request in response to a SAN administrator command that the displayed topology representation be refreshed. The requester, in turn, forwards the request to a request handler, for example, in a mark-up language format, such as XML, for further processing.

20

Further aspects of the invention provide a SAN as described above in which the interface element includes a manager daemon in communication with the request handler and the manager process, for example, via an object request broker. The request handler transmits the request to

36

IBM DOCKET NO. SJ09-2001-0096

the manager daemon which, in response, effects retrieval of the SAN representation from the manager process. The request handler can transmit the request to the manager daemon in the same format as that received from the requester. Alternatively, the request handler can map the request onto a generic format prior to its transmission to the manager daemon. The manager

5   daemon can, moreover, include a controller that receives the request from the request handler, and communicates with the manager process to retrieve the SAN representation.

In still further aspects, the invention provides a SAN as described above in which the user interface element includes a daemon process, herein referred to as user interface daemon, that

10   receives the SAN representation retrieved by the manager daemon. The user interface daemon, in turn, effects display of the SAN representation on the graphical output device.

Yet still further aspects of the invention provide a SAN as described above in which the manager daemon segregates a representation retrieved from the manager process, e.g., a SAN topology

15   representations, onto a plurality of sub-representation, and transmits the sub-representations to the user interface daemon.

Dynamically Extending File Systems

20   The invention provides, in other aspects, an improved SAN of type having one or more digital data processors, e.g., the aforementioned hosts, and one or more storage devices. At least a selected one of the hosts includes a file system that effects access by the host to assigned storage devices. In response to a request by (or on behalf of) the selected host for extension of the file

37

system, a manager assigns one of more further storage devices to that digital data processor. An agent associated with the first digital data processor that responds to the assignment by extending the file system to include the assigned storage device.

5   Further aspects of the invention provide a SAN as described above in which the agent automatically extends the selected host file system by executing one or more steps including initializing the assigned storage device, creating a logical object to represent the assigned storage device, adding the logical object into a logical grouping of storage devices that contain the file system to be extended, extending a volume size of the host file system, and increasing a size of

10   the host file system. In related aspects, the agent does not extend the file system if any of these steps fail.

Related aspects of the invention provide a SAN as described above in which the agent executes on an AIX journal system. Here, the agent extends the selected host file system by converting

15   the assigned storage device into one or more physical volumes, adding the one or more physical volume into a volume group of the file system to be extended, and extends the logical volume of that file system onto the assigned storage device.

Further related aspects invention provide a SAN as described above in which the agent executes

20   on a UNIX or Veritas file system (both running under a Solaris operating system). Here, the agent extends the selected host file system by writing a new label to the assigned storage device, configuring the storage device for use with a volume manager by converting the storage device into one or more VM disks, adding the one or more VM disks to a disk group where a logical

38